

PCS 5703 - Sistemas Multiagentes

JaCaMo

Henrique Donâncio

henriquedonancio@gmail.com

JaCaMo

- Jason: programação de agentes BDI utilizando um intérprete da linguagem *AgentSpeak(L)*
- CArtAgO: artefatos no modelo A & A (agentes e artefatos) *Java*
- MOISE+: organização do SMA *XML*

AgentSpeak(L)

- Um agente AgentSpeak(L) corresponde a especificação de um conjunto de crenças (base inicial do agente) e de um conjunto de planos (biblioteca de planos)
- Inspirada na arquitetura BDI
- Baseada no paradigma de programação lógica

Materiais:

Jason: Programming multi-agent systems in AgentSpeak using Jason. (livro)

3.4 Example: A Complete Agent Program (exemplo do livro)

<http://jason.sourceforge.net/>

CARTAGO: <http://cartago.sourceforge.net>

CARTAGO by Examples (exemplos disponíveis no site)

Moise+: <http://moise.sourceforge.net>

Tutorial integração JaCaMo: <https://github.com/hdonancio/jacamo>

Crenças

(i) baixo(pedro)

(ii) joga(pedro, basquete)

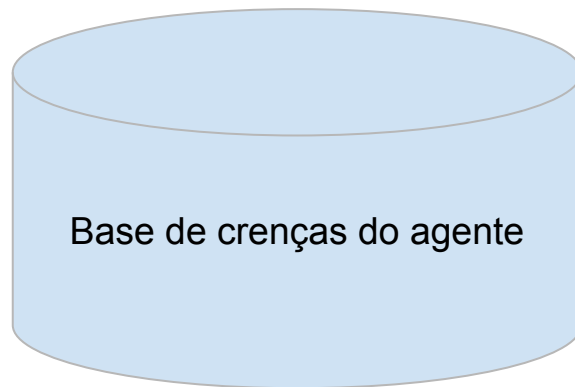
Adiciona crença: +joga(pedro, futebol).

Remove crença: -joga(pedro, basquete).

Atualiza crença: -+joga(pedro, rugby).

Consulta: ?joga(pedro, X) | ?joga(Y, basquete) | ?joga(pedro, Z, 2)

?data(X, agosto, 2017) | ?data(_, M, 2017) | ?data(D, M, 2017)



Objetivos

Denotados por “!”

!meta.

!contador(0).

!soma(X, Y).

Planos

(i) evento-ativador: contexto <- corpo

(ii) `#!start : true <- .print("Hello World").`

(iii) `#!preparar_jantar(Amigos): Amigos & lugares na mesa(N) & Amigos < N
<-`

(iv) `#!verifica : valor(X) & X < 10 <- .print("O valor é menor que 10.").`

Planos

```
/* Initial beliefs and rules */
```

```
valor(10).
```

```
/* Initial goals */
```

```
!verifica.
```

```
/* Plans */
```

```
+!verifica : valor(X) & X < 0 <- .print("O valor é menor que 0.").
```

```
+!verifica : valor(X) & X < 50 <- .print("O valor é menor que 50.").
```

```
+!verifica: valor(X) & X < 100 <- .print("O valor é menor que 100.").
```


Planos

```
/* Initial beliefs and rules */  
contador(0).
```

```
/* Initial goals */  
!start.
```

```
/* Plans */  
+!start : contador(X) & X < 10  
    <- .print("Iteração: ", X);  
    Y = X + 1;  
    -+contador(Y);  
!start.
```

```
+!start: true.
```

Regras

```
/* Initial beliefs and rules */
```

```
can_afford(Something) :- price(Something, P) & bank_balance(B) & B>P.
```

```
desire_items(car).
```

```
/* Initial goals */
```

```
!buy.
```

```
/* Plans */
```

```
+!buy: desire_items(Something) & can_afford(Something)
```

```
<- ...
```

“Bibliotecas”

Agentes podem servir de biblioteca de agentes!

```
{ include("common-rules.asl") }
```

```
{ include("common-plans.asl") }
```

```
{ include("actions.asl") }
```

Comunicação

`.send(destinatário, objetivo, mensagem)`

Objetivo (comunicativo):

tell (adiciona uma crença no agente destinatário)

achieve (desencadeia um plano no agente destinatário)

askOne/askAll (pergunta algo ao agente destinatário).

Exemplo comunicação

```
/* Agente 1
```

```
!iniciar.
```

```
+!iniciar: true <- .send(ag2, tell, ola).
```

```
/* Agente 2
```

```
!iniciar.
```

```
+!iniciar: true <- .print("Olá mundo").
```

```
+ola [source(Ag)]
```

```
<- .print("O ag ", Ag, " enviou um olá");
```

```
...
```

Ações internas

.print

.send

.broadcast

.drop_all_desires

.my_name

.concat

.length

.min

.max

.nth

.sort

.substring

.drop_all_events

.abolish

.string

.count

.create_agent

.date

.wait

.random

.kill_agent

.time

.perceive

.stopMAS

CArtAgO

- Framework independente da plataforma ou modelo de agente
- “Conecta” e/ou implementa recursos além da arquitetura de agentes
- Modela recursos do ambiente
- Artefatos (Classes Java) criados a nível de agentes
- *defineObsProperty , signal("tick"), OpFeedbackParam<Type>;*

Configurando o projeto (.mas2j)

MAS example {

environment:

c4jason.CartagoEnvironment

agents:

ag1 agentArchClass c4jason.CAgentArch;

classpath:

"../lib/cartago.jar"; //arcabouço cartago

"../lib/c4jason.jar"; //integração jason-cartago

//"../lib/moise.jar";

}

Artefato

```
import cartago.*;
```

```
public class Counter extends Artifact {
```

```
    void init(){  
        defineObsProperty("count",0);  
    }
```

```
    @OPERATION void inc(){  
        ObsProperty prop = getObsProperty("count");  
        prop.updateValue(prop.intValue()+1);  
        signal("tick");  
    }  
}
```

Artefato

```
!create_and_use.
```

```
+!create_and_use : true
```

```
<- !setupTool(lId);
```

```
  inc;
```

```
  inc [artifact_id(lId)].
```

```
+!setupTool(C): true
```

```
<- makeArtifact("c0", "c4jexamples.Counter", [], C).
```

MOISE+

Ferramenta com intuito de modelar a organização de um SMA:

- Dimensão estrutural: definem se papéis e ligações de heranças e grupos;
- Dimensão funcional: é estabelecido um conjunto de planos globais e missões para que as metas sejam atingidas;
- Dimensão deôntica: responsável pela definição de qual papel tem obrigação ou permissão para realizar cada missão

Exemplo MOISE+

```
<structural-specification>
<role-definitions>
  <role id="papel_A" />
  <role id="papel_B" />
  <role id="papel_C" />
</role-definitions>

<group-specification id="grupo_exemplo">
  <roles>
    <role id="papel_A" min="1" max="10" />
    <role id="papel_B" min="1" max="2" />
    <role id="papel_C" min="1" max="1" />
  </roles>
</group-specification>
</structural-specification>
```

!start.

+!start: true

```
<- makeArtifact("grupo_exemplo","ora4mas.nopl.GroupBoard",["  
src/exemplo-os.xml", grupo_exemplo, false, true],GrArtId);  
adoptRole(papel_C)[artifact_id(GroupBoard)];
```

MASSIM

URL: *<https://github.com/casals/mapc-base>*

JDK 8 (configurar variáveis de ambiente)

Eclipse Kepler

Porta 12300 definida por padrão em:

"conf/test-completescenario/2015-complete-3sims.xml"

Referências

BORDINI, R. H.; HÜBNER, J. F. JaCaMo Project. 2014. [Http://jacamo.sourceforge.net/](http://jacamo.sourceforge.net/)

BORDINI, R. H.; HÜBNER, J. F.; WOOLDRIDGE, M. Programming Multi-Agent Systems in AgentSpeak using Jason. New Jersey: Wiley, 2007.

BRATMAN, M. Intention, plans, and practical reason. [S.l.]: Harvard University Press, 1987.

HÜBNER, J. F. Um Modelo de Reorganização de Sistemas Multiagentes. Tese (Doutorado) — Universidade de São Paulo, São Paulo, 2003.

RAO, A. S. AgentSpeak(L): BDI agents speak out in a logical computable language. In: Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World. Berlin: Springer, 1996, (LNCS)

RICCI, A.; SANTI, A.; PIUNTI, M. CArtaGO (Common Atifact infrastructure for Agents Open environments). 2014. Disponível em: <[http://apice.unibo.it/xwiki/bin/view/ CARTAGO](http://apice.unibo.it/xwiki/bin/view/CARTAGO)>.

Tutorial JaCaMo, disponível em :<https://github.com/hdonancio/jacamo>